

**Proyecto Final**  
**Procesamiento Digital de Señales**  
**Guía de Laboratorio**

Cristian Camilo Martínez Escobar

**2023-1**

**Facultad de Ingeniería**  
**Universidad de Antioquia**

## **1. Análisis financiero del mercado de criptomonedas**

En esta práctica de laboratorio exploraremos cómo utilizar la biblioteca cctx para obtener datos de criptomonedas de una plataforma de intercambio (Exchange) almacenar esta información en un archivo .csv y luego aplicar un procesamiento a esta señal para obtener más información del mercado. Con esta combinación de herramientas, podremos analizar la tendencia del mercado en diferentes temporalidades y tomar decisiones en inversiones o análisis de criptomonedas.

En los últimos años, el mercado de las criptomonedas ha experimentado un crecimiento significativo y se ha convertido en un área de interés para muchos inversores y entusiastas. Con la ayuda de Python y sus bibliotecas, podemos acceder y analizar los datos de las criptomonedas de manera eficiente, lo que nos permite obtener información valiosa sobre su comportamiento.

### **1.1 Capturar los datos de las criptomonedas y almacenarlas en un archivo CSV**

Utilizaremos la biblioteca cctx, una biblioteca de intercambio de criptomonedas en Python que proporciona una interfaz unificada para interactuar con varios exchanges populares. Utilizaremos esta biblioteca para obtener datos de precios de criptomonedas en tiempo real desde un Exchange seleccionado.

Adjunto a la guía de laboratorio se proporciona un notebook de python llamado Obtener\_Datos\_Criptomonedas.ipynb, abra una ventana de Google Colab y abra este notebook modifique el código para generar 2 archivos en diferentes temporalidades de tiempo como se indica a continuación y descárguelos en la carpeta de trabajo de este laboratorio.

Si el ultimo digito de su Cedula es ...

( 1 6 3 ) ---> “ETH/USDT”

( 2 6 4 ) ---> “BTC/USDT”

( 5 6 7 ) ---> “LTC/USDT”

( 6 6 8 ) ---> “ADA/USDT”

( 9 6 0 ) ---> “TRX/USDT”

Genere un archivo con temporalidad de 1 semana “1w” y otro con temporalidad de “1h” de la criptomoneda que le corresponda.

**Hint:** Modifique la variable moneda y cambie el timeframe

```
moneda = 'BTC/USDT'  
bars = exchange.fetch_ohlcv(moneda, timeframe='1h', limit=1000)
```

## 1.2 Filtro de Media Móvil para analizar la tendencia del mercado.

El filtro de media movil (Moving Average, MA) es un filtro que permite eliminar ruido aleatorio de alta frecuencia. Este filtro tiene la siguiente ecuación en diferencias:

$$\frac{1}{L} \sum_{i=0}^{L-1} x[n-i]$$

Donde L es el número de puntos usados (orden del filtro),  $x[n]$  es la señal de entrada y  $y[n]$  es la señal de salida. La media móvil es un indicador que suaviza los datos de precios al promediar los valores en un período de tiempo específico.

### Indicador técnico usando cruce de Medias Móviles.

El cruce de medias móviles es una estrategia utilizada en el análisis técnico para identificar posibles cambios en la tendencia de un activo financiero. Esta estrategia se basa en el cruce de dos o más medias móviles de diferentes períodos. Al utilizar diferentes períodos para las medias móviles, se pueden obtener señales de compra o venta cuando estas líneas se cruzan entre sí.

El cruce de medias móviles más común es el cruce de una media móvil más corta con una media móvil más larga. Cuando la media móvil más corta cruza por encima de la media móvil más larga, se considera una señal de compra. Esto indica un posible cambio en la tendencia alcista del precio y sugiere que podría haber oportunidades de compra. Por otro lado, cuando la media móvil más corta cruza por debajo de la media móvil más larga, se interpreta como una señal de venta. Esto indica un posible cambio en la tendencia bajista del precio y sugiere que podría haber oportunidades de venta.

Una vez que tengamos los datos de las criptomonedas, aplicaremos un filtro de media móvil, una técnica de procesamiento de señales, que nos permitirá suavizar los datos y resaltar las tendencias del mercado en diferentes temporalidades.

1. Cargue el archivo CSV de temporalidad semanal "1w" generado anteriormente.
2. Diseñe una función que calcule un filtro de media móvil con parámetros de entrada los datos y la ventana que se desea usar en el filtro.
3. Ahora Suavice la información del precio de cierre de usando el filtro de media móvil con una ventana de 10, 55 y 100 y gráfíquelos en 2 subplots. Uno con la gráfica del precio y otro con el precio y todas las medias móviles.
4. Concluya cual es la tendencia del mercado en el corto plazo de tiempo es decir cómo se está moviendo la media móvil menor a comparación de las otras.

En el análisis del mercado de valores, lo más común es utilizar una media móvil de 50 o 200 días para ver las tendencias del mercado e indicar hacia dónde se dirigen los valores.

Lo ideal es que el tiempo entre mediciones sea el mismo y que sea suficiente para que la señal no varíe demasiado. Si la señal varía de forma continua y suave es posible que el filtro casi no afecte a la forma original de la señal. Sin embargo si hay cambios bruscos puede suavizar estos cambios.

### **1.3 Comparación de la función del filtro en diferentes temporalidades.**

1. Cargue el archivo CSV de temporalidad horaria "1h" generado anteriormente.
2. Suavice la información del precio de cierre de usando el filtro de media móvil con una ventana de 21 y 50 periodos, gráfíquelos en la misma grafica junto con el precio de cierre.
3. Compare el nivel de suavizado con el grafico anterior, ¿porque se debe esto?
4. cuál cree que es la mejor MA para filtrar el ruido de la volatilidad de estas criptomonedas en un gráfico horario.

## **2. Aplicación de filtros en cascada para procesar señales de audio**

En la industria musical y más específicamente en un equipo de mezcla de DJ, los ecualizadores permiten a los DJs ajustar y manipular el espectro de frecuencias de las pistas en tiempo real, mezclando diferentes canciones y logrando transiciones suaves y armoniosas. Al aplicar filtros pasa-bajas, pasa-bandas y pasa-altas, los DJs pueden controlar y resaltar diferentes elementos sonoros de las pistas, adaptándolas al estilo y la energía deseada en su mezcla.

La respuesta de frecuencia de un filtro describe cómo el filtro afecta las diferentes frecuencias en una señal. Puede ser representada gráficamente como un gráfico de amplitud o magnitud en función de la frecuencia, mostrando qué frecuencias son atenuadas o amplificadas por el filtro.

Realice un filtro pasa-bandas usando un filtro pasa-bajas y uno pasa-altas, el cual deje pasar un rango de frecuencias medias. ¿Qué consideraciones debe tener en cuenta con la frecuencia de corte de los filtros pasa-bajo y pasa-alto?, explique.

Grafique la respuesta en frecuencia de cada filtro y luego grafique la señal filtrada en el tiempo.

### 3. Análisis del espectrograma luego de aplicar filtros digitales respuesta del filtro.

Los filtros digitales son algoritmos utilizados para alterar el espectro de frecuencia de una señal. Pueden eliminar o atenuar ciertas frecuencias, cambiar la amplitud relativa de las frecuencias o aplicar efectos específicos a la señal. Los filtros digitales se implementan mediante técnicas de procesamiento de señales, como convolución y multiplicación en el dominio de la frecuencia.

Un espectrograma es una representación visual del espectro de frecuencia de una señal en función del tiempo. Muestra cómo varían las frecuencias a lo largo del tiempo y proporciona información sobre la energía o amplitud de diferentes componentes de frecuencia en la señal.

La idea de esta sección es analizar el espectrograma después de aplicar filtros digitales y obtener información sobre cómo se han modificado las características de frecuencia de la señal en el tiempo. Para esto haga uso de las siguientes funciones y grafique el espectrograma de la señal de audio y luego la señal filtrada con el pasa-banda del ítem anterior.

```
def extraer_ventanas(signal, size, fs):
    #Tamano de paso
    step=int(0.010*fs)
    n_seg = int((len(signal) - size) / step)
    # extraer segmentos
    windows = [signal[i * step : i * step + size]
                for i in range(n_seg)]
    # stack (cada fila es una ventana)
    return np.vstack(windows)

def potspec(X, size, n_padded_min=0):
    # Zero padding para la proxima potencia de 2
    if n_padded_min==0:
        n_padded = max(n_padded_min, int(2 ** np.ceil(np.log(size)
/ np.log(2))))
    else:
        n_padded = n_padded_min

    # Transformada de Fourier
    Y = np.fft.fft(X, n=n_padded)
    Y = np.absolute(Y)

    # non-redundant part
    m = int(n_padded / 2) + 1
    Y = Y[:, :m]
    return np.abs(Y) ** 2, n_padded
```

Repita el ítem numero **2** pero realice un Rechaza-Bandas utilizando 2 filtros consecutivamente, luego grafique el espectrograma de la señal filtrada.

¿Qué diferencia hay en las frecuencias de corte del filtro Pasa-Banda?, explique.

#### **4. Conclusiones**

Realice conclusiones generales sobre la práctica.